

IdeaExchange

De Jorge Casero Martín-Albo

Índice

1. Introducción
 - a. Presentación del proyecto
 - b. Objetivos del proyecto
 - c. Justificación del proyecto
2. Análisis de requerimientos
 - a. Identificación de necesidades y requerimientos
 - b. Identificación de público
 - c. Estudio de mercado y competencia
3. Diseño y planificación
 - a. Definición de la arquitectura del proyecto
 - b. Diseño de la interfaz de usuario
 - c. Planificación de las tareas y los recursos necesarios
4. Implementación y pruebas
 - a. Desarrollo de las funcionalidades del proyecto
 - b. Pruebas unitarias y de integración
 - c. Corrección de errores y optimización del rendimiento
5. Documentación
 - a. Documentación técnica
 - b. Documentación de usuario
 - c. Manual de instalación y configuración
6. Mantenimiento y evolución
 - a. Plan de mantenimiento y soporte
 - b. Identificación de posibles mejoras y evolución del proyecto
 - c. Actualizaciones y mejoras futuras
7. Conclusiones
 - a. Evaluación del proyecto
 - b. Cumplimiento de objetivos y requisitos
 - c. Lecciones aprendidas y recomendaciones para futuros proyectos
8. Bibliografía y referencias
 - a. Fuentes utilizadas en el proyecto
 - b. Referencias y enlaces de interés

1)Introducción

– Presentación del proyecto:

IdeaExchange es una página parecida a un foro cuyos usuarios registrados puedan interactuar y compartir código e ideas sobre programación. La plataforma ofrece una interfaz intuitiva que permite a los usuarios crear publicaciones, editarlas, eliminarlas, votarlas y también eliminar su cuenta. IdeaExchange se enfoca en la difusión de ideas, tanto propias como *públicas* y que se hayan encontrado online.

- Objetivos del proyecto:

Los objetivos principales de IdeaExchange son:

- Ofrecer a los usuarios un foro donde puedan interactuar e intercambiar ideas y código. Los usuarios pueden compartir sus conocimientos y experiencias con otros miembros.
- Simplificar la creación y gestión de publicaciones. Además, los usuarios tienen la opción de editar sus publicaciones para corregir errores o actualizar información. Esto garantiza un intercambio eficiente y preciso de ideas y conocimientos, y los administradores pueden verificar las publicaciones, notificar a otros usuarios que el código funciona y que la capacidad de los enlaces potenciales es válida.
- Permite a los usuarios votar las publicaciones de otros usuarios para evaluar su relevancia y calidad. La función de votación de IdeaExchange permite a los usuarios expresar sus opiniones sobre las publicaciones de otros usuarios. Los votos pueden ser positivos, neutrales o negativos, destacando los mensajes más valiosos y fomentando la participación activa de la comunidad en la evaluación del contenido.
- Proporcionar una estructura organizada para los temas y subtemas, lo que facilita la navegación y búsqueda de contenido específico. IdeaExchange permite una navegación clara y una búsqueda precisa gracias a la división y subdivisión de temas y subtemas. Los usuarios pueden explorar diferentes áreas de interés y acceder fácilmente al contenido relevante dentro de cada tema lo cual facilita la ubicación y el descubrimiento de contenido específico.
- Brindar a los usuarios la opción de eliminar su cuenta si así lo desean, ofreciendo control y privacidad: IdeaExchange valora la privacidad y el control de los usuarios sobre su propia información.

- Justificación del proyecto:

- La justificación de IdeaExchange se basa en la necesidad de un espacio virtual donde los usuarios puedan interactuar y compartir ideas de manera sencilla y estructurada. IdeaExchange no es la única aplicación con ese objetivo, pero sí que es la primera que busca una solución más casual y desenfadada, lejos de la seriedad que típicamente acompaña a la programación. Al tener una estructura jerárquica clara, los usuarios pueden explorar áreas de interés específicas y acceder rápidamente a la información que buscan. Esto mejora la experiencia del usuario al ahorrar tiempo y esfuerzo en la búsqueda de contenido específico.
- En resumen, IdeaExchange se justifica como un proyecto que satisface la necesidad de un foro moderno y funcional que promueva la colaboración y el intercambio de conocimientos, enfocado en el público que está interesado en la programación no sólo como trabajo, sino también o como hobby o que desea ponerse metas y compartir recursos.

2)Análisis de requerimientos

- Identificación de necesidades y requerimientos:

Requerimientos funcionales:

- Registro de usuarios: Los usuarios deben poder registrarse en la plataforma proporcionando información básica como nombre de usuario, correo electrónico y contraseña. Además, se debe implementar un sistema de validación de datos para garantizar la integridad de la información ingresada.
- Creación de publicaciones: Los usuarios registrados deben poder crear nuevas publicaciones en los subtemas correspondientes. La creación de publicaciones debe permitir la inclusión de contenido escrito (como puede ser el código o una explicación, así como enlaces a posibles repositorios externos o pequeñas demos) para enriquecer las publicaciones.
- Edición y eliminación de publicaciones: Los usuarios deben tener la opción de editar y eliminar las publicaciones que han creado. Esto les permitirá corregir errores, actualizar información o eliminar publicaciones que ya no sean relevantes.
- Votación de publicaciones: Los usuarios deben poder votar de forma positiva o negativa las publicaciones de otros usuarios, así como poder retirar el voto. Esta función ayudará a clasificar y destacar las publicaciones más valiosas y relevantes dentro de la comunidad.
- Eliminación de cuenta: Los usuarios deben tener la opción de eliminar su cuenta de forma rápida y total.

Requerimientos no funcionales:

- Base de datos SQL: Para almacenar de manera eficiente la información de los usuarios, publicaciones y votos, se utilizará una base de datos SQL. Se deben implementar consultas y transacciones adecuadas para garantizar la integridad y la consistencia de los datos.
- Backend con PHP: El backend de IdeaExchange se implementará utilizando PHP como lenguaje de programación. PHP se encargará de gestionar la lógica de negocio, procesar las solicitudes del frontend y comunicarse con la base de datos para almacenar y recuperar la información necesaria.

- Frontend con Angular: El frontend de IdeaExchange se desarrollará utilizando Angular, un framework de JavaScript ampliamente utilizado. Angular permitirá crear una single page application (SPA) que brinda una experiencia de usuario fluida e interactiva. Se utilizarán componentes, servicios y enrutamiento para crear una interfaz de usuario intuitiva y atractiva.
- Servidor local con XAMPP: Durante la fase de desarrollo, se utilizará XAMPP como servidor y base de datos local para ejecutar la aplicación. Esto permitirá realizar pruebas y depuración de manera eficiente antes de la implementación en un entorno de producción.

- Identificación de público:

- El público objetivo de IdeaExchange está compuesto por usuarios con diferentes niveles de conocimiento y con la programación fuera del entorno laboral como interés común. Estudiantes, profesionales, admiradores de la tecnología... En general cualquier persona interesada en aprender, compartir y colaborar. La plataforma está diseñada para ser accesible y fácil de usar, tanto para novatos como para aquellos con experiencia en el uso de foros.

- Estudio de mercado y competencia:

- Se ha realizado un estudio de mercado y una evaluación de la competencia para obtener información sobre las mejores prácticas y características que se encuentran en otras plataformas de foros. Se han identificado competidores que ofrecen funcionalidades y características similares, y se ha analizado cómo abordan las necesidades de los usuarios y brindan una experiencia satisfactoria.
- Existen muchas páginas web que ofrecen algo parecido, pero como ya se ha comentado anteriormente lo hacen desde una posición más seria, más imponente. Es por eso que he hallado este hueco, y por lo que he decidido crear IdeaExchange.

3) Diseño y planificación

- Definición de la arquitectura del proyecto:

La arquitectura del proyecto IdeaExchange se basa en una estructura cliente-servidor, donde el backend se implementa utilizando vanilla PHP y la base de datos SQL, mientras que el frontend se desarrolla utilizando Angular como framework de JavaScript.

La arquitectura general se puede describir de la siguiente manera:

- Cliente: El cliente es una single page application (SPA) desarrollada en Angular. Proporciona la interfaz de usuario interactiva y permite a los usuarios interactuar con el sistema, cómo crear publicaciones, votar, editar y eliminar publicaciones, y gestionar su cuenta.
- Servidor: El servidor se implementa utilizando PHP y se encarga de procesar las solicitudes del cliente, interactuar con la base de datos y proporcionar los datos necesarios al cliente en formato JSON.
- Base de datos: Se utiliza una base de datos SQL para almacenar la información del sistema, incluyendo datos de usuarios, publicaciones, votos y otros detalles relevantes. Se establecen relaciones adecuadas entre las tablas para mantener la integridad y la coherencia de los datos.
- Comunicación: La comunicación entre el cliente y el servidor se realiza a través de solicitudes HTTP utilizando el protocolo REST (Representational State Transfer). El cliente envía solicitudes al servidor, que procesa la solicitud y envía una respuesta en formato JSON.

- Diseño de la interfaz de usuario:

El diseño de la interfaz de usuario en IdeaExchange se centra en ofrecer una experiencia intuitiva y atractiva para los usuarios. Al ser una SPA, se busca una navegación fluida y una interacción sin interrupciones.

La interfaz de usuario contendrá los siguientes elementos principales:

- Barra de navegación lateral: Proporciona acceso a las diferentes secciones del foro
- Barra superior: donde el usuario podrá hacer login y/o registrarse.
- Lista de publicaciones: Muestra las publicaciones más recientes y populares, organizadas en función de los temas y subtemas correspondientes. Cada publicación mostrará información relevante, como el autor, la fecha de creación y la cantidad de votos.
- Formularios de creación y edición de publicaciones: Permiten a los usuarios crear nuevas publicaciones y editar las existentes. Estos formularios incluirán campos para ingresar contenido, enlaces o imágenes, y botones para guardar o cancelar las operaciones.
- Opciones de votación: Cada publicación tendrá opciones para que los usuarios voten de forma positiva, neutral o negativa, lo que ayudará a clasificar y resaltar las publicaciones más valiosas.
- Gestión de cuenta: Los usuarios podrán acceder a una sección donde podrán ver y editar su información de perfil, así como tener la opción de eliminar su cuenta si lo desean.

- Planificación de las tareas y los recursos necesarios:

- Análisis y diseño detallado: Comprender los requisitos del proyecto, diseñar la arquitectura, definir las interfaces y estudiar posibles interfaces de usuario.
- Desarrollo del backend: Implementar la conexión con la base de datos, desarrollar los controladores y modelos, garantizar la integridad de los datos.
- Desarrollo del frontend: Crear los componentes, servicios y enrutamiento necesarios, diseñar y desarrollar la interfaz de usuario, y buscar una experiencia de usuario fluida.
- Implementación de la base de datos: Configurar y administrar la base de datos SQL, diseñar las tablas y establecer las relaciones adecuadas.
- Pruebas y depuración: Realizar pruebas exhaustivas del sistema, identificar errores, y asegurarse de que todas las funcionalidades se comporten según lo esperado.
- Implementación y despliegue: Preparar el entorno de producción, configurar el servidor y la base de datos, y realizar el despliegue de la aplicación.

En cuanto a los recursos con los que desarrollar la aplicación, se requerirán:

- Programador con conocimientos en PHP, Angular, HTML, CSS y SQL.
- Entorno de desarrollo: todo el código ha sido escrito en Visual Studio Code, con PHPMyAdmin de XAMPP para la base de datos.
- Servidor de pruebas: el servidor escogido es Apache de XAMPP.
- Documentación: Documentar el proyecto, incluyendo manuales de usuario y guías de implementación.

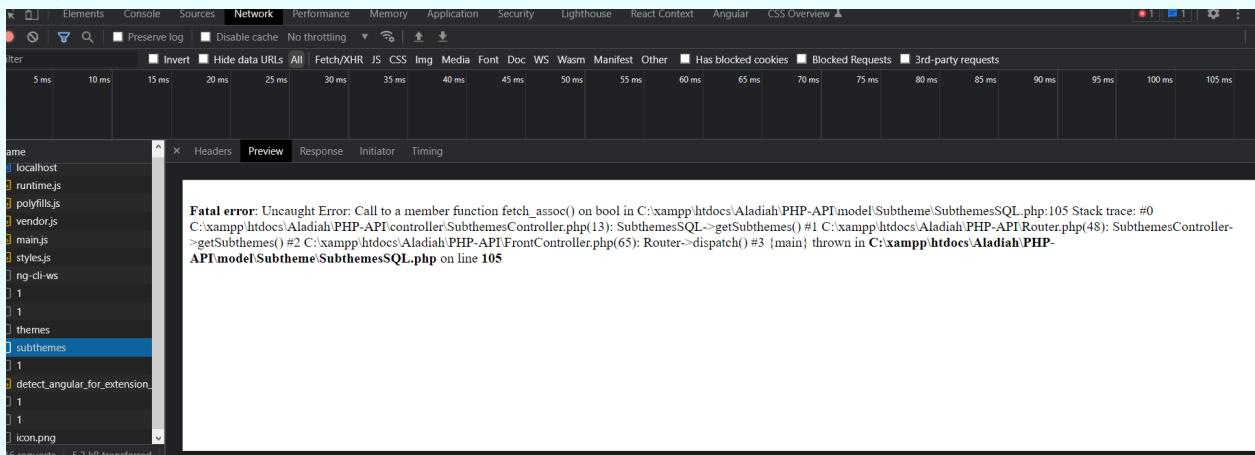
4) Implementación y pruebas

-Desarrollo de las funcionalidades del proyecto:

- Registro de usuarios: Los usuarios podrán registrarse en IdeaExchange proporcionando la información requerida, como nombre de usuario, dirección de correo electrónico y contraseña. Se verificará la validez de los datos ingresados y se almacenarán de manera segura en la base de datos. Además se busca añadir por comunicación asíncrona el comprobar que el nombre de usuario y la dirección de correo sean únicas, para evitar esa molestia que ocurre cuando todo parece estar bien, vas a registrarte y entonces te dice que revises los datos.
- Creación de publicaciones: Los usuarios registrados podrán crear nuevas publicaciones en los subtemas correspondientes. Se proporcionará un formulario donde se podrán ingresar el título, el contenido y cualquier otro detalle relevante. Las publicaciones se guardarán en la base de datos y se asociarán con el autor correspondiente.
- Edición y eliminación de publicaciones: Los usuarios podrán editar y eliminar las publicaciones que hayan creado. Se les permitirá realizar cambios en el contenido, actualizar detalles y eliminar la publicación si lo desean.
- Votación de publicaciones: Los usuarios podrán votar las publicaciones de otros usuarios de forma positiva, neutral o negativa. Se registrarán los votos y se calculará la puntuación total de cada publicación para mostrar su popularidad. Se implementarán medidas para evitar votos duplicados o fraudulentos.
- Gestión de cuentas: Los usuarios podrán acceder a una sección donde podrán ver y editar su información de perfil. También se proporcionará una opción para que los usuarios puedan eliminar su cuenta si así lo desean.

- Pruebas unitarias y de integración:

- Se realizarán pruebas exhaustivas para garantizar el correcto funcionamiento de las funcionalidades implementadas. Esto incluirá pruebas unitarias para cada componente y función individual, así como pruebas de integración para verificar la interacción correcta entre los componentes del sistema.
- Las pruebas unitarias se centrarán en comprobar que cada funcionalidad se comporte según lo esperado y cumpla con los requisitos establecidos. Las pruebas de integración se realizarán para asegurar que los diferentes componentes del sistema funcionen correctamente juntos. Se probarán escenarios de uso comunes, como la creación y edición de publicaciones, la votación de publicaciones y la gestión de cuentas.



- Aquí podemos encontrar un error a la hora de llamar a una de las APIs.

```

1 reference | 0 overrides
public function getSubthemes(): array {
    $sqlQuery = "SELECT `id-subtheme`, `main-theme-id`, `subtheme` FROM `subthemes`";
    if (!$statement = $this->conn->prepare($sqlQuery)) {
        die("Error when preparing the statement: " . $this->conn->error);
    }
    $result = $statement->execute();
    $subthemesArray = [];
    while ($row = $result->fetch_assoc()) {
        array_push($subthemesArray,
            new Subtheme(intval($row['id-subtheme']), intval($row['main-theme-id']),
                $row['subtheme']));
    }
    return $subthemesArray;
}
*/

```

```

1 reference | 0 overrides
public function getThemes(): array {
    $sqlQuery = "SELECT `id-theme`, `theme-name` FROM `themes`";
    if (!$statement = $this->conn->prepare($sqlQuery)) {
        die("Error when preparing the statement: " . $this->conn->error);
    }
    $reply = array();
    if($statement->execute()) {
        $result = $statement->get_result()->fetch_all();
        for($i = 0; $i<count($result); $i++) {
            array_push($reply, new Theme(
                intval($result[$i][0]),
                intval($result[$i][1]),
                $result[$i][2]
            ));
        }
    }
    return $reply;
}

```

- El error está en que getSubthemes aún tiene la primera implementación que hice, la cual no estaba bien implementada. Al lado la función get Themes, que es muy parecida pero su código está revisado.

```

1 reference | 0 overrides
public function getSubthemes(): array {
    $sqlQuery = "SELECT `id-subtheme`, `main-theme-id`, `subtheme` FROM `subthemes`";
    if (!$statement = $this->conn->prepare($sqlQuery)) {
        die("Error when preparing the statement: " . $this->conn->error);
    }
    $reply = array();
    if($statement->execute()) {
        $result = $statement->get_result()->fetch_all();
        for($i = 0; $i<count($result); $i++) {
            array_push($reply, new Subtheme(
                intval($result[$i][0]),
                intval($result[$i][1]),
                $result[$i][2]
            ));
        }
    }
    return $reply;
}
*/

```

```

1 reference | 0 overrides
public function getThemes(): array {
    $sqlQuery = "SELECT `id-theme`, `theme-name` FROM `themes`";
    if (!$statement = $this->conn->prepare($sqlQuery)) {
        die("Error when preparing the statement: " . $this->conn->error);
    }
    $reply = array();
    if($statement->execute()) {
        $result = $statement->get_result()->fetch_all();
        for($i = 0; $i<count($result); $i++) {
            array_push($reply, new Theme(
                intval($result[$i][0]),
                intval($result[$i][1])
            ));
        }
    }
    return $reply;
}

```

- Código revisado

Name	Headers	Preview	Response	Initiator	Timing
localhost			▼ [{idSubtheme: 1, subthemeName: "prueba", mainThemeId: 1},...]		
styles.css			► 0: {idSubtheme: 1, subthemeName: "prueba", mainThemeId: 1}		
runtime.js			► 1: {idSubtheme: 2, subthemeName: "HTML5", mainThemeId: 1}		
polyfills.js			► 2: {idSubtheme: 3, subthemeName: "HTML5", mainThemeId: 1}		
vendor.js			► 3: {idSubtheme: 4, subthemeName: "HTML5", mainThemeId: 1}		
main.js			► 4: {idSubtheme: 5, subthemeName: "HTML5", mainThemeId: 1}		
styles.js			► 5: {idSubtheme: 6, subthemeName: "HTML5", mainThemeId: 1}		
ng-cli-ws			► 6: {idSubtheme: 7, subthemeName: "HTML5", mainThemeId: 1}		
1			► 7: {idSubtheme: 8, subthemeName: "HTML5", mainThemeId: 1}		
1					
themes					
subthemes					
1					
detect.angular_for_extension_...					
icon.png					

- Demostración de que funciona

```

    name: 'HTML5',
    themeId: 1
}

this.themesService.deleteTheme(theme.id, 1).subscribe((e) => console.log(e));

navbarEnter(event: EventTarget | null): void {
  console.log("Enter");
  /* (<HTMLDivElement>event).classList; */
}

```

- Ahora vamos a comprobar el método deleteTheme. Como está en cascada en la base de datos, si funciona debería devolver un true y los subthemes deberían estar borrados.

Name	Headers	Preview	Response	Initiator	Timing
localhost	▼ General				
styles.css		Request URL: http://localhost/Aladiah/PHP-API/FrontController.php/theme/1/admin/1			
runtime.js		Request Method: DELETE			
polyfills.js		Status Code: 200 OK			
vendor.js		Remote Address: [::1]:80			
main.js		Referrer Policy: strict-origin-when-cross-origin			
styles.js	▼ Response Headers				
ng-cli-ws		Access-Control-Allow-Credentials: true			
1		Access-Control-Allow-Origin: http://localhost:4200			
1		Access-Control-Max-Age: 86400			
detect.angular_for_extension_...		Connection: Keep-Alive			
icon.png					

Name	X Headers	Preview	Response	Initiator	Timing
localhost					
styles.css		true			
runtime.js					
polyfills.js					
vendor.js					
main.js					
styles.js					
ng-cli-ws					
1					
1					
detect.angular_for_extension...					
icon.png					

- En efecto, devuelve un true. Ahora vamos a comprobar la base de datos

The screenshot shows the MySQL Workbench interface. On the left, the database tree displays various databases and tables, including 'themes' under 'qwelexandria'. The main pane shows a query results grid for the 'themes' table.

Query results:

```
SELECT * FROM `themes`
```

	id-theme	theme-name
<input type="checkbox"/>	2	HTML
<input type="checkbox"/>	3	HTML
<input type="checkbox"/>	4	HTML
<input type="checkbox"/>	5	HTML
<input type="checkbox"/>	6	HTML

Buttons at the bottom of the results grid:

- Check all
- With selected: Edit, Copy, Delete
- Export

pnpMyAdmin

Recent Favorites

New
anuncios_examen
cars
New
cars
concesionario
information_schema
mysql
performance_schema
phpmyadmin
pvpaint
qwelexandria
New
post-score
post-types
posts
subthemes
themes
users

Server: 127.0.0.1 » Database: qwelexandria » Table: subthemes

Browse Structure SQL Search Insert Export

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

SELECT * FROM `subthemes`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id-subtheme	main-theme-id	subtheme-name

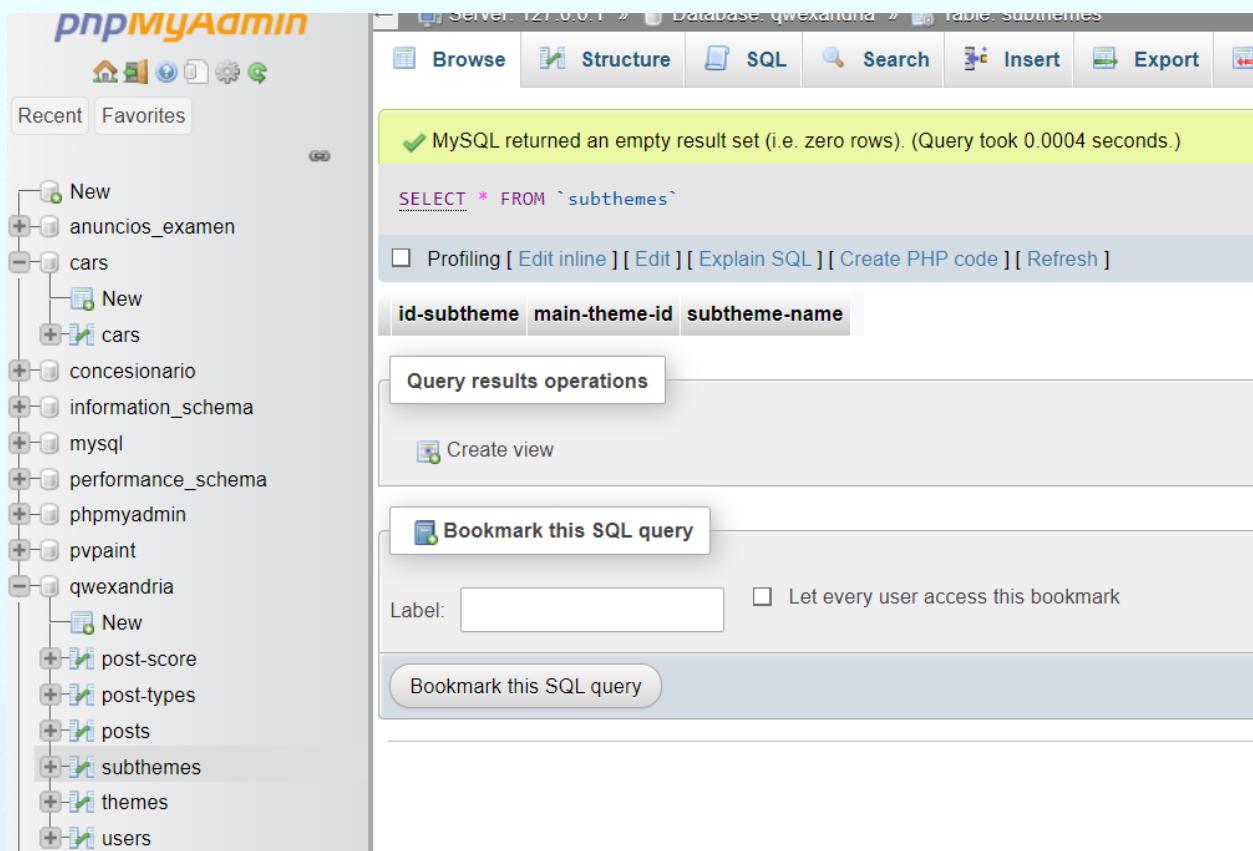
Query results operations

Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query



- Efectivamente, todos los subthemes han sido eliminados junto con el theme que tenía id 1

- Corrección de errores y optimización del rendimiento:

- Durante las pruebas, se identificarán errores y se corregirán para asegurar un funcionamiento sin problemas de IdeaExchange.
- Se recomienda realizar pruebas exhaustivas en diferentes navegadores web y con diferentes tamaños de datos para garantizar la estabilidad y el rendimiento de la aplicación en condiciones reales de uso.
- Una vez finalizada esta fase, se estará preparado para avanzar hacia la implementación y el despliegue en un entorno de producción.

5) Documentación

- Documentación técnica:

s PC > Local Disk (C:) > xampp > htdocs > Aladiah				
Name	Date modified	Type	Size	
Angular-App	29-May-23 14:44	File folder		
PHP-API	12-Jun-23 22:52	File folder		
api.htaccess	10-Jun-23 20:50	HTACCESS File	1 KB	
help.txt	12-Jun-23 01:04	Text Document	1 KB	
todo.txt	14-Jun-23 19:38	Text Document	1 KB	

- Arquitectura del sistema: Describe la estructura general del sistema, incluyendo la separación de frontend y backend, el uso de una base de datos SQL y la comunicación entre los diferentes componentes.

Se separará el frontend del backend, y el MVC o modelo-vista-controlador también se verá separado. En app (Angular-App) se encuentra la vista y las peticiones a las API REST, así como el manejo de los datos json, mientras que en api (PHP-API) se encuentran los modelos y controladores, encargados de relacionarse con la base de datos y enviar la información en formato json.

Dado que el proyecto está en desarrollo, la arquitectura podría cambiar un poco, como por ejemplo el ubicar el índice en el directorio raíz, así como limpiar los archivos .txt que son sólo usados durante el desarrollo.

- Tecnologías utilizadas:

Base de datos SQL: Al ser una base de datos relacional se mantiene la integridad de la información, mayoritariamente mediante el uso de primary y foreign keys.

Backend PHP: Ligero y fácil de usar, resulta muy rápido escribir una serie de REST API y poder encontrar, editarla y mantenerla.

Frontend: Angular es un framework de javascript que permite manipular la información y el manejo de la página de forma modular, lo que permite la separación de componentes.

Servidor: XAMPP, dado que es una herramienta todo en uno y evita muchos de los posibles problemas de compatibilidad que puedan surgir.

- Diseño de la base de datos: Incluye el esquema de la base de datos SQL utilizado en IdeaExchange, mostrando las tablas, las relaciones y los campos clave.
- Diseño del backend: El backend comienza en el archivo FrontController.php. Este registra todas las posibles acciones API gracias a una clase Router creada de manera propia, que se asegurará de que sólo se procesen las peticiones que se deseen y que lleven el método (GET, POST, DELETE...) necesario, así como una URL específica para cada API. Este Router después llevará cada petición a su controlador, que se comunicará con los modelos para efectuar las operaciones necesarias, devolviendo SIEMPRE un json al frontend para que este pueda actuar de manera acorde.
- Diseño del frontend: Al estar hecho en Angular, el frontend se separa en componentes y subcomponentes. Para empezar hay 3 componentes iniciales; la barra de navegación, que es la lateral, la barra de usuario, que es la superior y el espacio restante en el centro y esquina inferior izquierda de el navegador, que es donde se irá actualizando la información. Dentro de este componente existen los subcomponentes que se van a ir renderizando, cada uno con su propia URL totalmente y únicamente implementada en frontend, ya que se trata de una SPA o single page application.

- Documentación de usuario:

- Registro de usuarios: Los usuarios podrán registrar haciendo click en el botón register/registrarse en la barra de navegación superior, el cual abrirá una ventana de diálogo flotante en la cual ingresar los datos correspondientes. Una vez estén ingresados los usuarios deberán darle a registrar y, a continuación, podrán identificarse clicando en un botón en la misma barra de navegación, que de nuevo abrirá una ventana flotante en la que ingresar tus credenciales con un botón para hacer login.
- Navegación por el foro: La navegación es simple e intuitiva. La página principal tiene una lista con las publicaciones más valoradas, así como un filtro para buscar según subtemas. En el menú lateral hay una lista con temas y sus subtemas, y haciendo click en los subtemas te llevará a un listado con todos los subtemas relacionados.
- Creación y edición de publicaciones: Clicando en un botón llamado nueva publicación en la barra superior se abrirá una ventana flotante donde se podrá crear una nueva publicación. Debemos introducir el nombre de la publicación, una descripción y al menos un subtema al cual pertenece, el cual se elegirá mediante un desplegable. Después sólo tendremos que hacer click sobre publicar.

- Votación de publicaciones: Cada publicación tiene un botón con una flecha apuntando hacia arriba y otra apuntando hacia abajo. Clicando en cada una se vota de manera positiva o negativa, y clicando sobre la elección se borra el voto.
- Gestión de cuentas: Haciendo click sobre tu nombre de usuario en la parte izquierda de la barra de navegación superior podrás acceder a opciones de tu cuenta, como revisar los datos, modificarlos o incluso borrar tu cuenta.

- Manual de instalación y configuración:

- Instalación de dependencias: Para poder correr la aplicación de manera local en modo de desarrollo, es necesario descargar e instalar XAMPP. A continuación debemos descargar e instalar Nodejs, y utilizar los comandos pertinentes para instalar tanto Typescript como Angular.
- Configuración de la base de datos: Después de esto tendremos que iniciar el servidor Apache y la base de datos, e ingresar la base de datos. Importamos la base de datos que existe en el repositorio del proyecto.
- Configuración del servidor: Después, descargamos la carpeta del proyecto y buscamos en la raíz la carpeta htdocs, creada por XAMPP. Pegamos la carpeta del proyecto y ya; la configuración del servidor está en la carpeta (posiblemente oculta) “api.htaccess”.
- Despliegue de la aplicación: Navegamos hacia la carpeta “Angular-App” en la ventana de cmd y escribimos “ng serve”. Si todo va bien, se montará la página en local y podremos visitarla con el enlace local que el propio comando ng serve nos proporcione. Si ocurre algún error debería ser muy fácil identificarlo gracias a las herramientas que tienen los navegadores, ya bien sea el log de la consola, el log de network o un error a la hora de desplegar la aplicación.

6) Mantenimiento y evolución

- Plan de mantenimiento y soporte:

- Actualizaciones de seguridad: Debemos mantener actualizadas todas las tecnologías utilizadas en el proyecto, incluyendo el backend y el frontend. En otras palabras; aplicar parches de seguridad y actualizaciones periódicas para proteger contra posibles vulnerabilidades.
- Copias de seguridad regulares: Crear un respaldo de la base de datos en SQL con cierta periodicidad, al igual que controlar las modificaciones y versiones de la aplicación en Github, en caso de ser necesario obtener el código origen.
- Resolución de problemas y corrección de errores: Estar atento a cualquier imprevisto y error de la página que pueda poner en riesgo la integridad de la aplicación y de la información de los usuarios.
- Soporte técnico: Proporcionar un correo electrónico a través del cual los usuarios puedan reportar errores o comportamientos inusuales de la página.

- Identificación de posibles mejoras y evolución del proyecto:

- Interfaz de usuario: Añadir nuevas funcionalidades, como por ejemplo opciones de añadir una imagen de usuarios, chat o un tipo de mensajería y poder verificar la cuenta.
- Funcionalidades adicionales: Añadir una sección de comentarios, así como una forma de lograr que las actualizaciones de las publicaciones estén más resaltadas.
- Mejoras de rendimiento: Mejorar la eficiencia del código, la optimización de consultas de base de datos o la implementación de técnicas de almacenamiento en caché conforme se vayan encontrando errores, código redundante o se vaya aprendiendo más.
- Integraciones con otras plataformas: Evaluar la posibilidad de integrar IdeaExchange con otras plataformas o servicios populares, como redes sociales, sistemas de autenticación de terceros o sistemas de almacenamiento en la nube.

- Actualizaciones y mejoras futuras:

- Seguimiento de nuevas versiones de tecnologías: Mantenerse al tanto de las novedades de la tecnología usada, así como buscar unas posibles mejores alternativas a las que se usan; aprender un framework de PHP e implementarlo por ejemplo, o cambiar la base de datos a una que sea más útil a la hora de enlazar comentarios con usuarios, publicaciones, votos...
- Retroalimentación de los usuarios: Recopilar regularmente comentarios y sugerencias de los usuarios para identificar áreas de mejora y priorizar las actualizaciones y mejoras futuras.
- Planificación de lanzamientos: Crear un cronograma de lanzamientos para introducir nuevas características y mejoras. Esto permite mantener un ritmo constante de desarrollo y brindar a los usuarios nuevas funcionalidades.

7) Conclusiones

- Evaluación del proyecto:

El proyecto IdeaExchange se creó con la intención de desarrollar un foro donde los usuarios puedan interactuar, compartir ideas y participar en diversas acciones, como crear, editar y eliminar publicaciones, votar publicaciones y eliminar cuentas. Las características clave se han logrado implementar con éxito durante el proceso de desarrollo.

- Cumplimiento de objetivos y requisitos:

- En general, el proyecto IdeaExchange ha tenido éxito en el cumplimiento de los objetivos y especificaciones predeterminados. Los usuarios pueden registrarse, participar en las actividades del foro y administrar sus cuentas de manera efectiva. Las acciones de creación, edición y eliminación de publicaciones, así como la votación de posts, funcionan según lo esperado. Además, la organización de temas y subtemas permite una navegación sencilla y facilita el descubrimiento de información.

- Lecciones aprendidas y recomendaciones para futuros proyectos:

- Planificación adecuada: Una planificación y definición clara de los objetivos y requerimientos es esencial para garantizar un desarrollo eficiente.
- Pruebas exhaustivas: Realizar pruebas unitarias y de integración de manera rigurosa es crucial para garantizar el correcto funcionamiento de las funcionalidades y detectar posibles errores o problemas de rendimiento.

En cuanto a las recomendaciones para futuros proyectos similares, se sugiere:

- Establecer una comunicación efectiva con los usuarios y recopilar constantemente sus comentarios y sugerencias para mejorar la experiencia del usuario.
- Considerar la implementación de funcionalidades adicionales, como la capacidad de seguir a otros usuarios, recibir notificaciones o tener un sistema de mensajería interna, para mejorar la interacción y participación de los usuarios en el foro.

8) Bibliografía y referencias

- Fuentes utilizadas en el proyecto:

Durante el desarrollo de IdeaExchange se utilizaron diversas fuentes de información, incluyendo documentación oficial de las tecnologías utilizadas, foros de desarrollo y servidores de Discord, así como lo aprendido durante las prácticas y un chat de AI.

- Referencias y enlaces de interés:

- Tutoriales de REST API:

[Enlace 1](#)

[Enlace 2](#)

- Asistencia:

[Servidor de Discord "The Programmers Hangout"](#)

[Stack Overflow](#)

[Chat GPT](#)